

## The Dichotomy of SOA and ERP by Adam Jones

Despite efforts to enforce some level of IT governance, home-grown applications, small databases, and other day-to-day tools that make the job of your average person easier are bound to evolve.

### The Business Value of Information

Knowing everything about something is impossible. However, given enough time and labor, you can find out quite a bit about resources, personnel, equipment, and their attributes. That is the concept of ERP – take all your corporate information and knowledge and put it under a single flag. Your financial systems, HR, material planning, and other applications are moved to a single platform. This experience can be both painful and expensive, but yields unification through a singular application set. Now for the good news: there is another approach, one that does not involve 10,000 seats of a software package you will not get to use until it is rolled out in 5 or 6 years. Hot out of the gate, it is SOA – Service Oriented Architecture.

### MVC Model: A Simplified Understanding of SOA

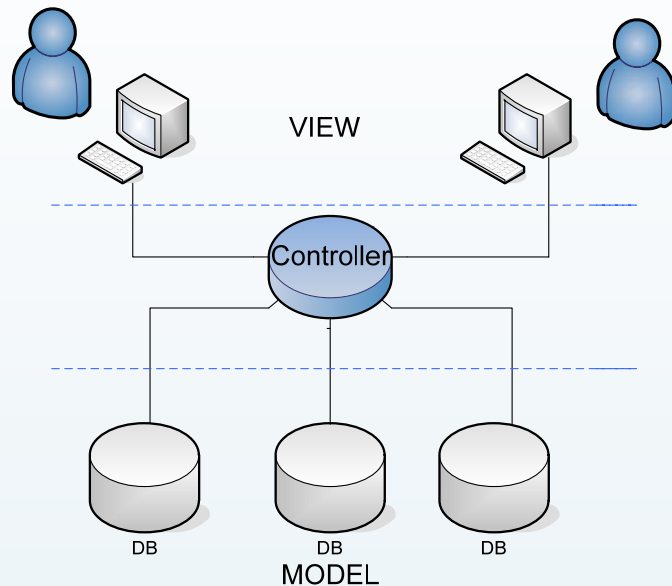
Service Oriented Architectures are essentially a **Model** (or data), a **View** (which represents the user), and a **Controller** that provide access to as many systems as is required, based around Web Services, a commu-

nication protocol that can be used to create web pages.

If you think of SOA as an architecture built in three layers, it becomes easier to understand. Let's start with how you get data: there are essentially three roles occurring when a user wants information. The first role

to communicate with the **Model**, then you are pretty much confined to a single type of **Model** (Oracle vs. MS Access vs. DB2, and so forth). What you really need in an environment with many different types of **Models**, such as the Navy Enterprise environment, is a **View** that can

### The MVC Model



is you – the user who wants to be able to view and manipulate the information, the **View**.

The next role to consider is the data itself – the information you want to access: the **Model**. To be clear, the **Model** can be a collection of disparate databases/file systems/other sources of data. Many **Models** have specific ways with which they like to be communicated. If your **View** only knows one way

display data without regard for the nature of the **Model** — a display that can report from any database.

To do this, a middle-tier layer called the **Controller** will be used. It will translate **Views** back and forth from **Models**, regardless of their type. For example, you can change the **View**, and all you have to do is tweak the Controller – the

**Model** will never know or care that the **View** has been altered or replaced. Likewise, you can switch the **Model**, then tweak the **Controller**, and your **View** will never know the difference.

To keep with the MVC paradigm, in the SOA model, the Controller is connected to the **View** using Web Services and each **Model** becomes service-enabled. Each Service becomes a small **Controller**, offering data to the **View**, gaining the ability to assemble custom **Views** (and even create custom applications) in a rapid, flexible way.

### So, What Do You Get Out of All This?

Offering users a single access point of information without involving them in the services implementation can help businesses and organizations respond more quickly and cost-effectively to changing market conditions. The need for SOA can arise in larger organizations with a huge number of systems or situations, such as mergers and acquisitions, where it allows two different companies to “smooth” the wide array of applications used by operations teams. The driving goal of Service Oriented Architectures is to achieve what is called Enterprise Application Integration. This means that across your Enterprise, applications can communicate with each other and exchange data without regard to the nature of the system because each application will expose one or more Services. Also, SOA is proven to be indispensable in harmonizing the goals of the business and their IT departments, to build up the infrastructure for long-term use in conjunction with the objectives of the business units, including the creation of custom applications that will display the results they need to achieve their tasks. SOA’s ability to customize the **View** enables everyone to make quicker, better, and more efficient decisions based on the aggregation of the data you need from any relevant system. This type of architecture is about providing the best information available to make the best decision possible.

### Final Challenges

Are Service Oriented Architectures going to revolutionize enterprises and government organiza-

tions? With a flexible architecture in hand, you can provide faster and more accurate access to organizational and external data. Enterprises have many useful applications, but they could never cover every conceivable need, because that was simply not possible. Now with SOA, when an unusual or custom request occurs, technologists can be responsive to it in a way that permits reuse, reconfiguration, or redesign with a minimum of effort, and yet still adheres to larger organizational objectives.

### About The Author

---

*Adam Jones, Director of Technical Services, performs Enterprise Architecture duties for one of the top acquisition programs in the Marine Corps and is a consultant to many engineering and logistics organizations within the Navy. He may be reached at [ajones@columbiagroup.com](mailto:ajones@columbiagroup.com) or (202) 546-1435 ext. 219*